

TRABAJO PRÁCTICO N°1

Ingeniería de Software

Plan de Gestión de las Configuraciones

Funes Joaquin
Pardina Lucas
Carpio Ramiro

Historial de revisiones

FECHA	OBSERVACIONES	AUTORES
09/04/2019	Armado inicial del PCM	Funes Joaquin, Pardina Lucas, Carpio Ramiro
10/04/2019	Continuación del armado del PCM	Funes Joaquin, Pardina Lucas, Carpio Ramiro

Indice de Contenidos

1. Introducción
 - 1.1 Propósito y alcance
 - 1.2 Propósito del Manejo de la Configuración del Software (SCM)
 - 1.3 Referencias, Abreviaturas y Glosario
 - 1.4 Conceptos Básicos
 - 1.5 Herramientas de Gestión de Configuración
2. Roles de la Administración de la Configuración
 - 2.1 Administración de la Configuración del Proyecto
 - 2.2 Esquema de Directorios
 - 2.3 Normas de Etiquetado y Versionado de Archivos
 - 2.4 Plan de Esquema de ramas
 - 2.5 Políticas de Fusión de Archivos y de etiquetado de acuerdo al progreso
 - 2.6 Forma de Entrega del Release
 - 2.7 Comité de Control de Cambios (CCB)
 - 2.8 Integrantes del equipo, roles y reuniones

1. Introducción

1.1 Propósito y alcance

Este documento trata sobre el Manejo del Plan de Configuración, la intención de este documento es describir la forma de trabajo, los documentos, el hardware, software y las herramientas utilizadas en este proyecto.

1.2 Propósito del Manejo de la Configuración del Software (SCM)

- Asegurar la consistencia de la información poniendo en práctica la SCM.
- Definir las personas que le dan soporte a las prácticas de SCM.
- Mantener la integridad a lo largo de todo el ciclo de vida del producto.
- Informar a grupos y las personas el estado del proyecto.
- Crear un historial del estado anterior y actual del proyecto.
- Mejora de Procesos.

1.3 Referencias, abreviaturas y glosario

ABREVIATURA	DESCRIPCIÓN
SCM	Gestión de la Configuración del Software (Software Configuration Management)
SVM	Sistema de control de versiones
CCB	Comité de Control de Cambios
SCI	Software Configuration item

SCMer	Rol encargado de realizar la tarea de gestión de la configuración (SCM)
-------	---

1.4 Conceptos Básicos

CI: Ítem de configuración, son todos los elementos creados durante el proceso. Pueden ser de 3 tipos:

1. De software: código fuente, recursos gráficos, bibliotecas, ejecutables.
2. Documentos: técnicos, administrativos y de usuario.
3. Estructura de datos: estructura de base de datos, datos iniciales, archivos de configuración, etc.

Versión: Toda evolución de un ítem de configuración.

Etiqueta: Las etiquetas son identificadores pegados a algunas versiones de determinados ítems de configuración.

Ramas: Las ramas son caminos independientes de evolución de las versiones de los ítems de configuración que facilitan el desarrollo en paralelo.

Repositorio: Espacio físico (directorio y grupo de directorios), donde se almacenan, una vez terminados, todos los elementos generados durante el proceso de desarrollo en sus diferentes versiones. Estos elementos pueden estar en tres estados: pendientes de aprobar por el área de SQA, aprobados por SQA, y en producción.

Check Out: Tomar un SCI del repositorio y copiarlo en un área de trabajo, dejando bloqueado el SCI en el repositorio, para que nadie más pueda hacer un check out del mismo.

Check In: Si se trata de un SCI nuevo, ingresarlo en el repositorio y dejarlo disponible para posteriores modificaciones, mediante el mecanismo de check out y check in. Si se trata de un SCI existente al que se le hizo check out, el check in implica el ingreso al repositorio de una nueva versión del SCI, dejando al SCI disponible, liberando el bloqueo que registrado en el momento del check out.

Release: Entrega de un sistema que se libera para su uso a los clientes.

1.5 Herramientas de Gestión de Configuración

REFERENCIA	PROPÓSITO	LINK
Java	Lenguaje de programación	www.java.com
NetBeans IDE	Entorno de desarrollo	www.netbeans.org
Git	Software de manejo de versiones	www.git-scm.com
GitHub	Sistema de control de versiones	https://github.com/joafunes/ingsoft
Travis	Herramienta de integración continua	https://travis-ci.org/joafunes/ingsoft
Gradle	Herramienta de automatización	www.gradle.org

2. Roles de la Administración de la Configuración

2.1 Administración de la Configuración del Proyecto

Las actividades de la gestión Configuraciones del proyecto serán coordinadas por el Gerente de Configuraciones Global (GPCM) del mismo, rol que será asignado a una persona. Esta será responsable de actividades como seguimiento de las herramientas, creación de las nuevas ramas, creación del reléase, etc. Además, se dispondrá un GPCM de respaldo dentro del equipo.

La siguiente tabla muestra las personas que tomarán el rol de GPCM:

Rol	Responsable Principal	Responsable
-----	-----------------------	-------------

GLOBAL PCM	Joaquín Funes	Ramiro Caprio
------------	---------------	---------------

Responsabilidades:

Rol	Responsabilidad
GPCM	<p>Será responsable de todos los CI:</p> <ul style="list-style-type: none"> • Creación de las ramas y administración de sus políticas. • Etiquetado y lanzamiento de ramas. • Coordinar actividades del CM en el proyecto. • Asegurar la correcta ejecución del esquema del CM • Participar en las auditorías. • Analizar todas las novedades relacionadas al CM.
TPCM	<ul style="list-style-type: none"> • Asistir en la creación de etiquetas y ramas. • Creación de actividades para el equipo específico en las ramas. • Garantizar la integridad del producto y el seguimiento de los elementos de configuración propios del equipo. • Participación en las auditorías. • Analizar todas las novedades relacionadas al CM.
Team	<ul style="list-style-type: none"> • Ayudar a resolver conflictos durante la actividad de merge. • Asegurarse que los criterios de calidad de los entregables a la rama principal se cumplan. • Seguir todos los procesos asociados, política y prácticas definidas por sus roles asignados.

2.2 Esquema de directorios




En el siguiente link se encuentra nuestro repositorio del proyecto:

<https://github.com/joafunes/ingsoft/>

En dicho repositorio, el código fuente se encuentra:

[ingsoft / TP-Software-RC1 / src /](#)

Allí tenemos los directorios de todas las clases necesarias para el proyecto:

 <code>tp/software/rc1</code>
 <code>Admin.java</code>
 <code>Autent.form</code>
 <code>Autent.java</code>
 <code>HistorialMedico.java</code>
 <code>Main.java</code>
 <code>Medico.java</code>
 <code>NuevoPaciente.form</code>
 <code>NuevoPaciente.java</code>
 <code>Paciente.java</code>
 <code>PrincipalAdmin.form</code>
 <code>PrincipalAdmin.java</code>
 <code>PrincipalDoctor.form</code>
 <code>PrincipalDoctor.java</code>
 <code>Turnos.java</code>
 <code>Usuario.java</code>

2.3 Normas de etiquetado y de nombramiento de los archivos

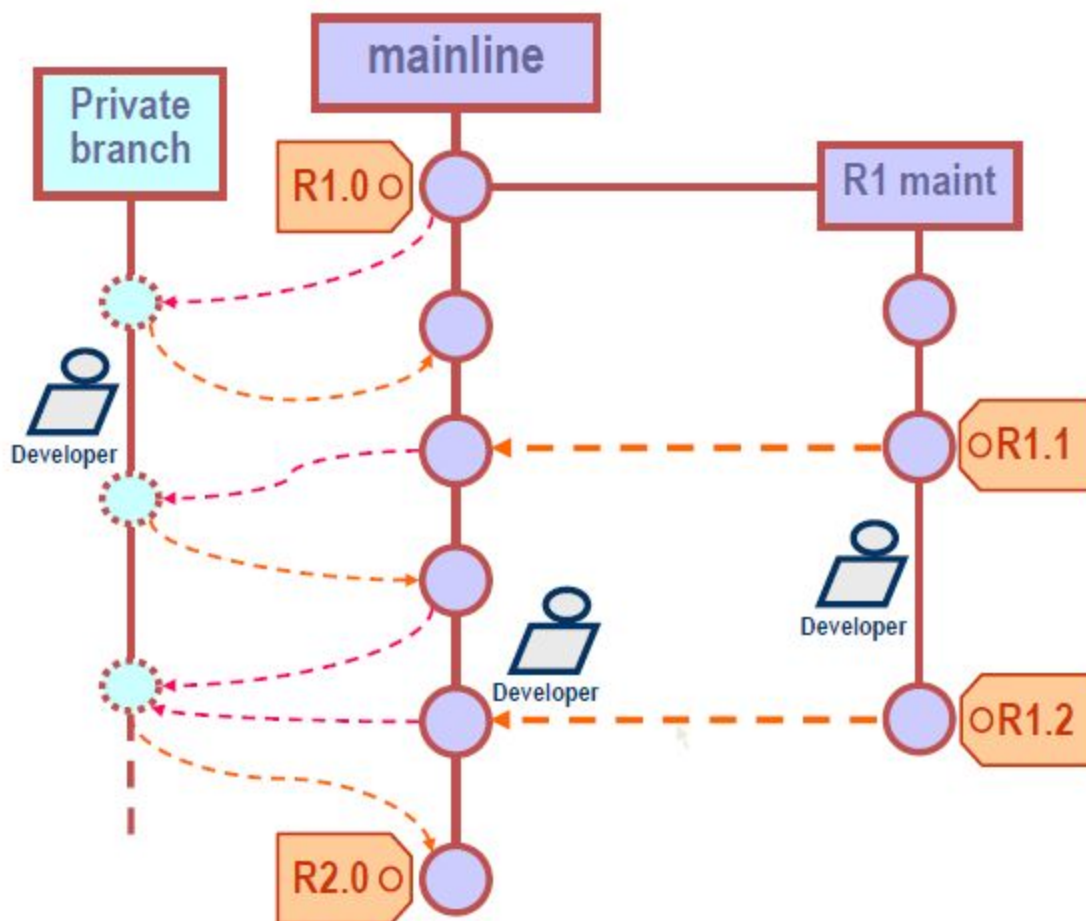
Para el nombramiento de etiquetas se seguirá una notación numérica compuesta por 3 números separados por puntos con la siguiente notación:

X.Y.Z

- X: se le conoce como versión mayor e indica la versión principal del software.
- Y: se le conoce como versión menor e indica nuevas funcionalidades.
- Z: se le conoce como revisión y nos indica que se hizo una revisión del código por algún fallo.

2.4 Plan de esquema de ramas a usar

El esquema de ramas a usar será “Lean and Lazy”. Para evitar que se repitan tareas en las distintas líneas, cada branch debe representar el desarrollo una tarea concreta para que se vuelva a integrar en la línea principal de desarrollo en poco tiempo, de esta manera la tarea de integración será menos costosa.



2.5 Políticas de fusión de archivos y de etiquetado de acuerdo al progreso

Para fusionar nuevas ramas que pudieran surgir del código principal, el administrador realizará la fusión haciendo un rebase del código junto a los desarrolladores. De esta forma rápidamente se podrán identificar errores, los desarrolladores podrán trabajar en resolverlos y el administrador estará al tanto de la situación.

Los criterios para modificar (incrementar) cada uno de los contadores de la etiqueta de versión son los siguientes:

- **X:** nuevas funcionalidades claves del sistema respecto a la versión anterior debido a la inclusión de nuevos requerimientos o la inclusión de nuevos módulos.
- **Y:** cambios significativos en la forma en la que se ofrece una funcionalidad existente o corrección de grandes fallos del sistema.
- **Z:** se modifica por cada entrega de software que se realice, que podría ser para corregir un pequeño fallo.

2.6 Forma de entrega del Release

Una vez terminado el código y finalizadas las pruebas, se procede a la compilación donde el compilador generará un archivo de extensión .jar , dicho archivo puede ser ejecutado ubicándolo en cualquier carpeta de la computadora del usuario, con la aclaración que debe estar instalado en el equipo cliente el sistema Java Runtime Enviroment (JRE). El código vendrá con su correspondiente documentación.

El software puede ser distribuido por medio de un sitio web.

2.7 Comité de Control de Cambios (CCB)

El CCB (por sus siglas en inglés Change Control Board) es el comité que considera el efecto del cambio desde un punto de vista estratégico y organizacional, y deciden si el cambio en cuestión está justificado para poder autorizarlo antes de su implementación.

El CCB puede estar conformado por una o varias personas, pero es importante que éstas tengan una visión global del proyecto. La decisión del CCB y sus apreciaciones se registran en los apartados correspondientes del formulario del cambio.

Si el CCB aprueba el cambio, se genera la asignación de la tarea de llevar a cabo dicho

cambio. Esta asignación y el formulario de cambio correspondientes, llegarán al SCMer. Mientras que las peticiones de cambio rechazadas se cierran y ya no se emprenden más acciones.

El formulario de cambio debe incluir:

- Cambio requerido
- Solicitante del cambio
- Costos estimados del cambio
- Componentes afectados
- Prioridad del cambio
- Tiempo estimado para realizar el cambio
- Fecha de solicitud del cambio, de la aprobación y de la implementación
- Pruebas y reporta de pruebas

2.8 Integrantes del equipo, roles y reuniones

Los roles de los miembros del equipo son los siguientes:

- Engineering Manager
- Release Manager
- Scrum Team
- GPCM

Al ser 3 integrantes en el grupo, todos realizamos diversas tareas que engloban la mayoría de las responsabilidades de los roles nombrados anteriormente.

En cuanto a las reuniones, algunas fueron virtuales donde cada uno trabajaba desde su casa editando el documento y también tuvimos 2 reuniones reales de unas 3 hs cada una.

2.9 Herramienta de seguimiento de BUGS

Utilizamos la herramienta Issues proporcionada por GitHub para hacer seguimiento de los errores.